

動的プリファレンスを扱う議論フレームワークの 解集合プログラミングによる議論意味論の計算

稲葉 遼*1

若木 利子*2

芝浦工業大学 システム工学部 電子情報システム学科*3

1 はじめに

人間の行う議論をコンピュータネットワーク上のマルチエージェントにシミュレートさせるため、Dung[2]により議論フレームワーク (以後、*AF*) とその議論意味論が提案された。法廷の議論においては、新法優先や上位法優先等のプリファレンス自身を推論や議論の対象とする動的プリファレンスが存在し、それを扱った議論が行われている。2011年、動的プリファレンスを扱うよう坂間・井上の *PLP* 言語 [5] を拡張した階層的 *PLP* (P, Ψ, Φ) が若木 [9] により提案され、更に階層的 *PLP* から構築される非抽象動的プリファレンス付き議論フレームワーク (以後、*dynamic PAF*) と *P-extensions* によって与えられる議論意味論が提案された。本研究では、*dynamic PAF* の *P-extensions* と解集合が一対一に対応する一般論理プログラム (*NLP*) を提案し、解集合プログラミングによりプリファレンスを議論の対象としながら所与の議題の議論判定を行うことのできる議論計算エンジンを開発した。

2 準備

2.1 拡張論理プログラムと解集合プログラミング

定義 1 (拡張論理プログラム (ELP)) *ELP* は (1) の形式のルール集合である。 L, L_i はリテラル (アトム A または $\neg A$)、*not* は失敗による否定 (*NAF*) である。(1) の形式のルール r について、 L を頭部といい $head(r)$ で表す。

$$L \leftarrow L_1, \dots, L_m, not\ L_{m+1}, \dots, not\ L_n. \quad (1)$$

ELP の意味論は解集合により与えられる。*NLP* はルールに論理否定 $\neg$ が出現しない *ELP* のサブクラスである。 Lit_P を P の言語の基礎リテラルの集合と定義する。

ASP (Answer Set Programming) [3] は問題の知識を *ELP* 等の論理プログラムで表現し、その解集合から、それと一対一に対応する問題の解を得るプログラミング手法である。本研究で用いる *ASP* のクラスは *ELP* である。

2.2 Dung の議論意味論

Dung の 4 種類の意味論 [2] (complete, stable, preferred, grounded) は、抽象議論フレームワーク (*AF*) の下で定義される *extension* によって与えられ、それによって勝利する論証を判定する。一方、*ELP* P で記述されたエージェントの知識より非抽象議論フレームワーク AF_P が構成され [6, 8, 2]、その意味論は、 AF_P に関する Dung の議論意味論、即ち *extension* により与えられる。

定義 2 (論証 [6]) 所与の *ELP* P において、論証 Ag はルール $r_i \in P$ ($1 \leq i \leq n$) の本体の任意のリテラル L_j に対し、 $head(r_k) = L_j$ なるルール r_k ($k > i$) が存在するような P の基礎ルールの有限系列 $Ag = [r_1; \dots; r_n]$ で定義される。 $head(r_i)$ を Ag の結論という。特に $head(r_1)$ を論証 Ag の主張と言い、 $claim(Ag)$ で表す。 P から構成した極小論証の集合を $Args_P$ で表す。論証集合 E に対して $claims(E)$ を以下で定義する。

$$claims(E) \stackrel{\text{def}}{=} \{L \mid \text{任意の } Ag \in E \text{ に対し } L = claim(Ag)\}$$

定義 3 (非抽象議論フレームワーク) 攻撃の概念 [6] (rebut, undercut, attack, defeat 等) により構成された $Args_P$ 上の関係を $Attack_{SP}$ で表す。非抽象議論フレームワーク AF_P は $(Args_P, attack_{SP})$ で定義される (詳細は [8] 参照)。

3 dynamic PAF と議論意味論

階層的 *PLP* で表現されたエージェントの知識ベースより構成される *dynamic PAF* の議論意味論は *Sname P-extensions* により与えられ、それにより議論で勝利する論証を判定する。以下で *Sname* $\in \{\text{comepelte, stable, preferred, grounded}\}$ とする。

3.1 階層的 PLP [9]

N をルール名集合とする。 $\preceq (n_{r1}, n_{r2}, n_\delta)$ は priority atom と称され、「ルール名 $n_\delta \in N$ を持つルール δ により、「ルール名 $n_{r2} \in N$ のルール $r2$ がルール名 $n_{r1} \in N$ のルール $r1$ より高い優先度を持つ」ことを意味する。なお δ は、 $r1, r2$ のルールより 1 レベル上位のメタルールであり、 $r1, r2$ は同じメタレベルとする。

定義 4 (階層的 PLP) 階層的 *PLP* は (P, Ψ, Φ) で定義される。但し、 P は名前無しルール (1) と名前 $n_r \in N_0$ を持つ (2) の形式の *ELP* からなる。 Ψ はルール名が n_δ 、頭部が priority atom である (3) の形のルールからなる *ELP* である。

$$n_r : L \leftarrow L_1, \dots, L_m, not\ L_{m+1}, \dots, not\ L_n. \quad (2)$$

$$n_\delta : \preceq (n1, n2, n_\delta) \leftarrow L_1, \dots, L_m, not\ L_{m+1}, \dots, not\ L_n. \quad (3)$$

Φ は優先度 [9] の集合で、以下で定義されるリテラル集合 $\mathcal{L}$ 上の関係 $\preceq$ として定義され、反射的、推移的な擬順序 (preorder) である。

$$\mathcal{L} \stackrel{\text{def}}{=} Lit_P \setminus \{L \mid \text{名前 } n_r \text{ の } r \in P \text{ に対し } L = head(r)\} \cup N_0$$

3.2 dynamic PAF の形式化

エージェントの知識ベースが階層的 *PLP* (P, Ψ, Φ) で記述されている時、動的プリファレンスを議論で扱う *dynamic PAF* の議論フレームワーク $PAF(P, \Psi, \Phi)$ 、及び、*P-extension* で与えられる議論意味論を述べる。

定義 5 (dynamic PAF [9]) 階層的 *PLP* (P, Ψ, Φ) より、

$$\pi \stackrel{\text{def}}{=} P \cup \Psi \cup \Pi \cup \Delta$$

を定義する。但し Π, Δ は以下の *NLP* である。

$$\begin{aligned} \Pi : & \prec (X, Y) \leftarrow \prec (X, Y, Z), \prec (Y, X, U), X \neq Y, \\ & \quad not\ \prec (Y, X), not\ \prec (Z, U). \\ & \prec (X, Y) \leftarrow \prec (X, Y, Z), \prec (Y, X, U), \\ & \quad ng(Y, X) \leftarrow \prec (Y, X, U). \\ & \prec (X, Z, t(U, V)) \leftarrow \prec (X, Y, U), \prec (Y, Z, V). \\ \Delta : & n_r \leftarrow body(r), \text{名前 } n_r \in N \text{ の } r \in P \cup \Psi \text{ に対し} \end{aligned}$$

$\mathcal{E}_\pi$ を π から構成した $AF_\pi = (Args_\pi, attacks_\pi)$ の *Sname extensions* の集合とする。 $\prec$ は $(\Phi_s \cup \Phi_e)^*$ で定義され、推移的、非反射的な狭義の (半) 順序である。ここで Φ_s, Φ_e は以下で定義される。 $S|_X \stackrel{\text{def}}{=} S \cap X$, $E \in \mathcal{E}_\pi$ とする。

$$\begin{aligned} \Phi_s & \stackrel{\text{def}}{=} \{(e, e') \mid (e, e') \in \Phi^* \wedge (e', e) \notin \Phi^*\} \\ \Phi_e & \stackrel{\text{def}}{=} \bigcup_{E \in \mathcal{E}_\pi} claims(E)|_\Theta, \text{ 但し } \Theta = \{\prec (n, n') \mid n, n' \in N\} \end{aligned}$$

階層的 *PLP* (P, Ψ, Φ) から構成される *dynamic PAF* は、 $PAF(P, \Psi, \Phi)$ として以下で定義される。

$$PAF(P, \Psi, \Phi) \stackrel{\text{def}}{=} (Args_\pi, attacks_\pi, \prec)$$

但し、 $\prec$ は $Args_\pi$ 上の関係で、以下の関係式に基づき、 Φ の情報から自動生成される狭義の (半) 順序である。

$$Ag_1 < Ag_2 \text{ iff } claim(Ag_1) \prec claim(Ag_2)$$

定義 6 (プリファレンス関係 $\sqsubseteq$) 階層的 *PLP* (P, Ψ, Φ) から構成された $PAF(P, \Psi, \Phi)$ において、 $\mathcal{E}_\pi$ を $AF_\pi = (Args_\pi, Attacks_\pi)$ の *Sname extensions* の集合とする。 $\mathcal{E}_\pi$ 上の関係 $\sqsubseteq$ は以下で定義される。任意の $E_1, E_2, E_3 \in \mathcal{E}_\pi$ に対し、

- $E_1 \sqsubseteq E_1$
- $E_1 \sqsubseteq E_2$ とは、ある論証 $a_2 \in E_2 \setminus E_1$ に対し、 $a_1 > a_2$ なる $a_1 \in E_1 \setminus E_2$ が存在し、 $a_2 < a_3$ なる $a_3 \in E_1 \setminus E_2$ が存在しない
- $E_1 \sqsubseteq E_2$ かつ $E_2 \sqsubseteq E_3$ のとき $E_1 \sqsubseteq E_3$

$E_1 \sqsubseteq E_2$ のとき、 E_2 は E_1 よりも高い選好を持つと言う。 $\sqsubseteq$ は 1, 3 の項目より擬順序である。

定義 7 (*P-extension*) $\mathcal{E}_\pi$ を $AF_\pi = (Args_\pi, attacks_\pi)$ の *Sname extensions* の集合とする。*Sname extension* $E \in \mathcal{E}_\pi$ について、任意の $E' \in \mathcal{E}_\pi$ に対し、 $E \sqsubseteq E'$ ならば $E' \sqsubseteq E$ が成り立つ時、そしてその時に限り *extension* E は $PAF(P, \Psi, \Phi) = (Args_\pi, attacks_\pi, \prec)$ の *Sname P-extension* である。

4 ASP による PAF 意味論の計算

本研究では *dynamic PAF* の *P-extensions* と解集合が一対一に対応する変換論理プログラムを提案する。以後、

$$\mathcal{I} \stackrel{\text{def}}{=} \{in(a) \mid a \text{ は論証 } Ag \in Args_\pi \text{ の論証名}\} \text{ とする。}$$

定義 8 (Dung 意味論の計算) 階層的 *PLP* (P, Ψ, Φ) か

Computing argumentation semantics capable of handling dynamic preferences in answer set programming

*1 Ryo Inaba

*2 Toshiko Wakaki

*3 Shibaura Institute of Technology

ら構成された $\pi = P \cup \Psi \cup \Pi \cup \Delta$ に関する AF_π を $(Args_\pi, attacks_\pi)$ とする。以下の Θ_{comp} は文献 [7] に基づき *complete extensions* を, Θ_{st} は *stable extensions* を計算するための *NLP* である。

$\Theta_{comp} \stackrel{\text{def}}{=} \Theta_{AF} \cup \Theta_{Lab}$
$\Theta_{st} \stackrel{\text{def}}{=} \Theta_{AF} \cup \Theta_{Lab} \cup \{\leftarrow undec(X)\}$
Θ_{AF} : 論証 $\forall Ag \in Args_\pi$ の論証名 a に対し, $ag(a) \leftarrow$, 攻撃の $\forall (a, b) \in attacks_\pi$ に対し, $attack(a, b) \leftarrow$, Θ_{Lab} : $in(X) \leftarrow ag(X), not\ ng(X)$, $ng(X) \leftarrow in(Y), attack(Y, X)$, $ng(X) \leftarrow undec(Y), attack(Y, X)$, $out(X) \leftarrow in(Y), attack(Y, X)$, $undec(X) \leftarrow ag(X), not\ in(X), not\ out(X)$.

AF_π の *complete* と *stable extensions* は, 以下の定理 1 に基づいて計算される。以下で関数 *name* を用いて論証 Ag の名前を $name(Ag)$ で表す。

定理 1 (健全性/完全性定理 [7]) AF_π において, E が *complete (stable) extension* ならば, $S|_I = \{in(a) \mid \text{論証 } Ag \in E \text{ について } a = name(Ag)\}$ なる Θ_{comp} (Θ_{st}) の解集合 S が存在する。逆に, S が Θ_{comp} (Θ_{st}) の解集合ならば, $E = \{Ag \mid Ag \text{ は } in(a) \in S \text{ なる名前 } a = name(Ag) \text{ を持つ論証}\}$ なる AF_π の *complete (stable) extension* E が存在する。

以下の変換論理プログラム $tr[P, \Psi, \Phi; Sname]$ では $Sname = stable$ の場合, Θ は Θ_{st} , それ以外は Θ_{comp} を表す。文献 [7] と同様, 全単射関数 ψ と 2 引数述語記号 m_2 を用いて, $in(a) \in S_j$ s.t. $\psi(S_j) = j$ iff $m_2(a, j) \in M$ (1) が成り立つように Θ の任意の解集合 S_j ($1 \leq j \leq \xi$, 但し ξ は Θ の解集合の総数) の情報を m_2 と ext_no の述語記号の原子式を頭部を持つ Γ のルールにより, 変換論理プログラムの解集合 M に埋め込んでいる。

定義 9 (変換論理プログラム) $Sname$ 意味論の下で階層的 $PLP(P, \Psi, \Phi)$ から構成される dynamic PAF の変換論理プログラム $tr[P, \Psi, \Phi; Sname]$ は以下で定義される *NLP* である。但し, $\pi \stackrel{\text{def}}{=} P \cup \Psi \cup \Pi \cup \Delta$ 。

$tr[P, \Psi, \Phi; compelte] \stackrel{\text{def}}{=} \Theta \cup \Gamma \cup \Lambda'$
$tr[P, \Psi, \Phi; stable] \stackrel{\text{def}}{=} \Theta \cup \Gamma \cup \Lambda'$
$tr[P, \Psi, \Phi; preferred] \stackrel{\text{def}}{=} \Theta \cup \Gamma \cup \Xi_{pr} \cup \Lambda$
$tr[P, \Psi, \Phi; grounded] \stackrel{\text{def}}{=} \Theta \cup \Gamma \cup \Xi_{gr}$
ここで, $\Lambda = \Lambda_1 \cup \Lambda_2$, $\Lambda' = \Lambda \setminus \{1 \sim 3 \text{ のルール}\}$
$\Xi_{gr} \stackrel{\text{def}}{=} \Xi \cup \{\leftarrow c(N), not\ d(N)\}$, $\Xi_{pr} \stackrel{\text{def}}{=} \Xi \cup \{\leftarrow d(N), not\ c(N)\}$
Γ : $\forall a \in S_j \mathcal{I}$ に対し, $m_2(a, j) \leftarrow . ext_no(j) \leftarrow .$
Ξ : $c(N) \leftarrow ext_no(N), in(X), not\ m_2(X, N)$, $d(N) \leftarrow m_2(X, N), not\ in(X)$,
Λ_1 : $\forall Ag \in Args_\pi$ の論証名 a , $claim(Ag) = L$ に対し, $arg_claim(a, L_t) \leftarrow$, Δ の $\forall Ag \in Args_\pi$ の論証名 a , $claim(Ag) = L$ に対し, $arg_claim(a, n) \leftarrow$, $\forall Ag \in Args_\pi$ の論証名 a , Ag の主張 $\prec (n_1, n_2)$ に対し, $\prec_d (n_1, n_2, a) \leftarrow$, 優先度 $\forall (L_1, L_2) \in \Phi$ に対し, $\prec_s (L_1, L_2) \leftarrow$, 論証 $\forall Ag \in Args_\pi$ の論証名 a , $claim(Ag) = L$ に対し, $L \leftarrow in(a)$,
Λ_2 : 1. $c_2(N_1, N_2) \leftarrow ext_no(N_1), m_2(X, N_2), not\ m_2(X, N_1)$, 2. $d_2(N_1, N_2) \leftarrow ext_no(N_2), m_2(X, N_1), not\ m_2(X, N_2)$, 3. $non_pref(N_1) \leftarrow d_2(N_2, N_1), not\ c_2(N_2, N_1)$, 4. $preferred(N_1) \leftarrow ext_no(N), not\ non_pref(N)$, 5. $credulous(X) \leftarrow preferred(N), in(X), not\ m_2(X, N)$, 6. $skeptical(X) \leftarrow in(X), not\ credulous(X)$, 7. $\prec_{skp} (X, Y) \leftarrow skeptical(Z), \prec_d (X, Y, Z)$, 8. $\prec_{skp} (X, Y) \leftarrow \prec_s (X, Y), not\ \prec_s (Y, X)$, 9. $\prec_{skp} (X, Z) \leftarrow \prec_{skp} (X, Y), \prec_{skp} (Y, Z)$, 10. $\prec (X, Y) \leftarrow \prec_{skp} (V, W), arg_claim(X, V), arg_claim(Y, W)$, 11. $gr1(N, Y) \leftarrow in(Y), m_2(X, N), \prec (X, Y), not\ in(X), not\ m_2(Y, N)$, 12. $gr2(N, Y) \leftarrow in(Y), m_2(Z, N), \prec (Y, Z), not\ in(Z), not\ m_2(Y, N)$, 13. $\sqsubseteq (N) \leftarrow gr1(N, Y), not\ gr2(N, Y)$, 14. $rgr1(N, Y) \leftarrow in(X), m_2(Y, N), \prec (X, Y), not\ in(Y), not\ m_2(X, N)$, 15. $rgr2(N, Y) \leftarrow in(Z), m_2(Y, N), \prec (Y, Z), not\ in(Y), not\ m_2(Z, N)$, 16. $\sqsubseteq (N) \leftarrow rgr1(N, Y), not\ rgr2(N, Y)$, 17. $grt1(N_1, N_2, Y) \leftarrow ext_no(N_1), ext_no(N_2), m_2(X, N_1)$, $m_2(Y, N_2), \prec (X, Y), not\ m_2(Y, N_1), not\ m_2(X, N_2)$, 18. $grt2(N_1, N_2, Y) \leftarrow ext_no(N_1), ext_no(N_2), m_2(Z, N_1)$, $m_2(Y, N_2), \prec (Y, Z), not\ m_2(Y, N_1), not\ m_2(Z, N_2)$, 19. $\sqsubseteq (N_1, N_2) \leftarrow grt1(N_1, N_2, Y), not\ grt2(N_1, N_2, Y)$, 20. $preferred(N_1) \leftarrow preferred(N_2)$, 21. $\sqsubseteq (N_1, N_3) \leftarrow \sqsubseteq (N_1, N_2), \sqsubseteq (N_2, N_3)$, 22. $\sqsubseteq (N_2) \leftarrow \sqsubseteq (N_1), \sqsubseteq (N_2, N_1)$, 23. $\sqsubseteq (N_2) \leftarrow \sqsubseteq (N_1), \sqsubseteq (N_1, N_2)$, 24. $\leftarrow \sqsubseteq (N)$,

$tr[P, \Psi, \Phi; preferred]$ について説明する。 $tr[P, \Psi, \Phi; preferred]$ の解集合を M とする。文献 [7] より, $AF_\pi = (Args_\pi, attacks_\pi)$ の *complete* (及び *preferred*) *extension* E は, Θ (及び $\Theta \cup \Gamma \cup \Xi_{pr}$) の解集合 S から $E = \{Ag \mid Ag \in Args_\pi \text{ は } in(a) \in S \text{ なる名前 } a = name(Ag) \text{ を持つ論証}\}$ として得られる。他方, $\Lambda \setminus \{26 \text{ のルール}\}$ は層状プログラムであるので M から構成される $E_M = \{a \mid Ag \in Args_\pi \text{ は } in(a) \in M \text{ なる名前 } a = name(Ag) \text{ を持つ論証}\}$ は AF_π の *preferred extension* となる。ところで (1) により, Γ のルールより, Θ の任意の解集合 S_j ($1 \leq j \leq \xi$, 但し ξ は π の解集合の総数) の情報が m_2 等のアトムとして, $tr[P, \Psi, \Phi; preferred]$ に埋め込まれている。ここで各 S_j は $E_j = \{a \mid in(a) \in S_j\}$ なる AF_π の (abstract) *complete extension* を表すが Λ の 1~4 のルールより, $preferred(N) \in M$ であれば M に埋め込まれた S_N ($1 \leq N \leq \xi$ に対する E_N は *preferred extension* となる。ルール 5 により $credulous(X) \in M$ なる論証 (名) X が存在すれば, E_M の論証 $X \in E_M$ について $X \notin E_N$ なる *preferred extension* E_N が存在することを意味し, ルール 6 により $skeptical(X) \in M$ なる論証 (名) X について全ての E_N で $X \in E_N$ であり AF_π で skeptically に正当化された論証であることがわかる。ルール 7~9 のルールより $\prec = (\Phi_s \cup \Phi_e)^*$ が, ルール 10 より $Args_\pi$ 上の関係 $\prec$ が算出される。 Λ_2 の残りのルールで定義 6, 7 を計算している。

dynamic $PAF(P, \Psi, \Phi)$ の *preferred P-extension* は, 下記定理に基づき解集合プログラミングにより計算される。他の意味論についても同様の定理が成り立つ。

定理 2 (健全性/完全性) E が階層的 $PLP(P, \Phi)$ から構成した dynamic $PAF(P, \Psi, \Phi)$ の *preferred P-extension* ならば, 変換論理プログラム $tr[P, \Psi, \Phi; preferred]$ に $M|_I = \{in(a) \mid a = name(Ag) \text{ なる論証 } Ag \in E \text{ の名前}\}$ なる解集合 M が存在する。逆に, M が dynamic $PAF(P, \Psi, \Phi)$ から構成した $tr[P, \Psi, \Phi; preferred]$ の解集合ならば, $E = \{Ag \mid Ag \text{ は名前 } in(a) \in M \text{ なる名前 } a = name(Ag) \text{ を持つ論証}\}$ なる dynamic $PAF(P, \Psi, \Phi)$ の *preferred P-extension* E が存在する。

5 実装と評価

入力 P, Ψ, Φ から AF を構成し, 変換論理プログラム tr を記述して, ASP ソルバにより議論計算をするプログラムを C で実装した。以下の例題 [4] を用いて, 実装したプログラムの評価を行った。

P: BBC は晴れと予報しているので, 今日ロンドン乾燥する。
 Q: CNN は雨と予報しているので, 今日ロンドンは乾燥しない。
 P: しかし, BBC は CNN よりも信頼できる。
 Q: 一方で, 統計では CNN は BBC よりも正確に予報している。
 Q: そして, 統計に基づく比較というのは, あなたの直感に基づいた信頼の度合いよりも合理的なものである。

P: $bbc: dry(london) \leftarrow forecast(bbc, fine), not \neg dry(london)$,
 $cnn: \neg dry(london) \leftarrow forecast(cnn, rain), not dry(london)$,
 $forecast(bbc, fine), forecast(cnn, rain)$.
 Ψ : $\prec (cnn, bbc, trust), \prec (bbc, cnn, statics)$,
 $\prec (trust, statics, rational)$.

実装したプログラムの出力は以下となり, $\neg dry(london)$ が含まれることから, 「今日ロンドンは乾燥しない」という人間の思考と同じ議論判定結果が得られた。

```
ryo ~ bash — 65x5
$ ./comp-paf weather.txt
{a0, a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a12, a15, a16, a19}
with claims {¬dry(london), cnn, preceq_d(cnn,bbc,trust), preceq_d(bbc,cnn,statics), preceq_d(trust,statics,rational), forecast(bbc,fine), forecast(cnn,rain), trust, statics, rational}
```

6 おわりに

提案した変換論理プログラムの計算量の解析や、種々の問題での有用性の評価をすることが今後の課題である。

参考文献

- [1] M. Caminada: On the issue of reinstatement in argumentation, Proc. of JELIA06, pp. 111-123, LNAI No.4160, Springer, 2006.
- [2] P. M. Dung: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artificial Intelligence, 77, pp.321-357, 1995.
- [3] M. Gelfond, V. Lifschitz: Classical Negation in Logic Programs and Disjunctive Databases, New Generation Computing 9, pp.365-385, 1991.
- [4] S. Modgil: Reasoning about preferences in argumentation frameworks, Artificial Intelligence, 173, pp.901-934, 2009.
- [5] C. Sakama, K. Inoue: Prioritized logic programming and its application to commonsense reasoning, Artificial Intelligence, 123, pp.185-222, 2000.
- [6] R. Schweimeier, M. Schroeder: A Parameterised Hierarchy of Argumentation Semantics for Extended Logic Programming and its Application to the Well-founded Semantics, TPLP 5 (1-2), pp.207-242, 2005.
- [7] T. Wakaki, K. Nitta: Computing Argumentation Semantics in Answer Set Programming, New Frontiers in Artificial Intelligence, LNAI Vol. 5447, pp. 254-269, Springer-Verlag, 2008.
- [8] T. Wakaki: Preference-Based Argumentation Capturing Prioritized Logic Programming, Proc. of ArgMAS2010, Revised Selected and Invited Papers, LNAI, Vol. 6614, Springer, pp. 306-325, 2011.
- [9] T. Wakaki: Preference-Based Argumentation Handling Dynamic Preferences Built on Prioritized Logic Programming, Proc. of PRIMA-2011, LNAI, Vol. 7047, pp. 336-348, Springer (2011)